به نام خدا

آزمایشگاه سخت‌افزار

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نام، نام خانوادگی و شماره دانشجویی اعضای گروه:

فرزام زهدی‌نسب ـ ۹۷۱۰۵۹۹۶

دالیا داودی ـ ۹۷۱۱۰۳۹۳

علی بالاپور ـ ۹۷۱۰۱۳۲۶

فهرست مطالب

۱ مقدمه پروژه ۲
۱.۱ چکیده . . . ۲
۲.۱ مقدمه . . . ۲
۳.۱ چالش‌ها . . . ۴
۴.۱ کاربردها . . . ۵

۲ تحقیقات مقدماتی ۶
۱.۲ تحقیق درباره نوع قطعات بورد اصلی و بورد ریموت‌کنترلر . . . ۶
۱.۱.۲ برد اصلی پهپاد . . . ۶
۲.۱.۲ لینک‌های مفید . . . ۱۰
۲.۲ تحقیق راجع به نسخه اندروید مورد استفاده در ثابت‌افزار . . . ۱۱
۳.۲ تحقیق راجع به فایل‌سیستم اندروید . . . ۱۱
۴.۲ بررسی و جستجوی انجمن‌ها، فروم‌ها، و ریپازیتوری مرتبط با پهپادهای DJI و بررسی ابزارهای مختلف . . . ۱۲

۳ شبیه‌ساز ۱۳
۱.۳ تحقیق در مورد شبیه‌سازهای پردازنده ARM . . . ۱۳
۲.۳ بررسی و استفاده از QEMU برای شبیه‌سازی سیستم‌عامل‌های مختلف . . . ۱۳
۱.۲.۳ آزمایش ۱: شبیه‌سازی اندروید 8.1 نسخه پردازنده x86 . . . ۱۴
۲.۲.۳ آزمایش ۲: شبیه‌سازی rpios lite نسخه پردازنده ARM . . . ۱۵
۳.۲.۳ آزمایش ۳: شبیه‌سازی اندروید 4.4 نسخه پردازنده x86 . . . ۱۷

۴ اجرا ۱۹
۱.۴ دیکامپایل کردن برنامه موجود در ثابت‌افزار و بدست‌آوردن کد اندروید . . . ۱۹
۲.۴ تحلیل firmware با ابزار firmwalk . . . ۱۹
۳.۴ اجرای ثابت‌افزار مربوط به کنترلر بر روی QEMU . . . ۲۰
۱.۳.۴ ساختار کلی فایل نصبی اندروید . . . ۲۰
۲.۳.۴ نحوه ایجاد فایل system.sfs . . . ۲۰
۳.۳.۴ آزمایش ۱: شبیه‌سازی اندروید 4.4 نسخه پردازنده x86 . . . ۲۱
۴.۳.۴ آزمایش ۲: شبیه‌سازی نسخه دستکاری شده اندروید 4.4 با استفاده از فایل‌های دامپ . . . ۲۱
۵.۳.۴ آزمایش ۳: شبیه‌سازی نسخه دستکاری شده اندروید 4.4 با استفاده از ترکیب فایل‌های دامپ و اندروید . . . ۲۲

۵ نتیجه‌گیری ۲۴

۱ مقدمه پروژه

۱.۱ چکیده

در این پروژه قصد داریم تا ثابت‌افزار (Firmware) پهپاد Mavic dji ۲ را بررسی و تحلیل نماییم. روند کلی کار به این صورت است که در ابتدا این ثابت‌افزار بر روی شبیه‌ساز پردازنده ARM اجرا می‌شود و سپس اپلیکیشن‌هایی که در آن تعبیه شده را به‌طور دقیق بررسی شده، و هدف، نحوه کارکرد، سورس کد، روتین‌ها و ماژول‌های این اپلیکیشن‌ها را استخراج می‌گردد.
در این گزارش در ابتدا اطلاعات کلی در مورد خود پهپاد و کنترلر و کاربردهای آن و همچنین سخت‌افزار استفاده شده بحث می شود. سپس در بخش دوم در باره QEMU و نحوه اجرای ثابت افزار بحث می‌شود و همچنین گزارش تعدادی آزمایش برای آشنایی با نحوه استفاده از QEMU قرار داده شده است.

۲.۱ مقدمه

با توجه به پیشرفت‌های قابل توجه تکنولوژی در سال‌های اخیر مخصوصا در زمینه‌های تصویربرداری، پروتکل‌های ارتباطی، سیستم‌های نهفته، و ارتباط از راه دور، پهپادها نقش قابل توجهی در عرصه‌های مختلف (عکس‌برداری هوایی، مقاصد نظامی و ...) یافته‌اند. به طوریکه امروزه به پهپادها به چشم ابزارهای پیچیده جمع‌آوری اطلاعات در عرصه‌های مختلف نگاه می‌شود که می‌توانند یکی از ماده‌های اولیه نیاز اساسی بشر امروزی، یعنی داده خام را جمع‌آوری کنند. با استفاده از این داده‌های خام، می‌توان اطلاعات (information) و دانش (knowledge) استخراج کرد و گام‌های مهمی در جهت پیشرفت کسب‌وکار هدف برداشت.
امروزه از پهپاد استفاده‌های بسیاری می‌شود. به طور کلی به برخی از کاربردهای پهپادهای تجاری (commercial drones) در حوزه‌های مختلف می‌پردازیم:

- **محافظت از طبیعت**: داده‌های جمع آوری شده توسط پهپادها از مناطق طبیعی می‌توانند به متخصصان طبیعت در جهت حفاظت از منطقه مدنظر کمک کند. برای مثال، یک تیم تحقیقاتی محافظت از اقیانوس‌ها از پلتفرم DJI برای جمع‌آوری داده برای مانیتور کردن وضعیت سلامت اقیانوس‌ها استفاده می‌کند. استفاده از این پلتفرم منجر به جمع آوری سریع، دقیق و ارزان داده می‌شود.
- **صنعت کشاورزی**: با استفاده از پهپادها و دوربین‌های مخصوص، می‌توان زمین‌های کشاورزی را به‌صورت بهتر و دقیق‌تر مانیتور و بررسی کرد و یا می‌توان از پهپاد جهت سم‌پاشی استفاده کرد. کاربردهای پهپادها در این زمینه به‌گونه‌ای زیاد است که شرکت مطرح DJI یک پلتفرم اختصاصی برای این حوزه توسعه داده است.
- **تامین امنیت**: از پهپادها می‌توان جهت پایش و مانیتور کردن مکان‌های مهم (مانند مکان‌های نظامی) استفاده کرد و در صورت به تشخیص ناهنجاری (با استفاده از تکنیک‌های anomaly detection)، اقدامات لازم انجام شود. مزیت پهپادها نسبت به دوربین‌های مداربسته، قابلیت جابجایی بیشتر آن‌ها و همچنین دوربین‌های قوی‌تر می‌باشد. پهپاد DJI Matrice 300 RTK برای این منظور ساخته شده است.

- **صنعت معدن**: از پهپادهای مخصوص برای بررسی ساختار داخلی معادن و تونل‌ها استفاده می‌شود. این نوع پهپادها دارای محافظ مخصوص می‌باشند و دارای حسگرهای منحصربه‌فرد جهت پیدا کردن در زمان قطع ارتباط، هستند. همچنین این نوع پهپادها دارای تجهیزات ارتباطی بسیار قوی می‌باشند و می‌توانند از اعماق زمین نیز ارتباط با مرکز کنترل را حفظ کنند. پهپاد Elios 2 از این نوع می‌باشد.

- **فیلم‌برداری و تصویربرداری**: می‌توان گفت که پرکاربردترین کارکرد پهپادهای تجاری، فیلم‌برداری و تصویربرداری است. با استفاده از پهپادها می‌توان تصاویر خیره کننده‌ای را ثبت کرد. همچنین امروزه در بخش فیلم‌برداری اکثر فیلم‌های سینمایی بزرگ، از پهپادها استفاده می‌گردد. سری Mavic شرکت DJI یکی از بهترین پهپادها برای این کارکرد می‌باشند.

- **نقشه‌برداری**: با استفاده از نوع خاصی از پهپاد، می‌توان از یک منطقه جغرافیایی، نقشه‌برداری کرد. این پهپاد یک سری عکس از ارتفاع مخصوص از سطح زمین تهیه می‌کنند و سپس با استفاده از یک سیستم هوشمند و ایجاد یک مدل سه بعدی، به نقشه‌برداری می‌پردازند. پهپاد WingtraOne GEN II یکی از بهترین پهپادها در این زمینه است.

- **جمع‌آوری داده**: با توجه به مجهز بودن اکثر پهپادها به دوربین با کیفیت و سایز کوچک پهپاد، از آن‌ها برای جمع آوری تصاویر استفاده می‌شود. با توجه به نیاز روزافزون تکنولوژی‌های جدید مانند هوش مصنوعی و یادگیری عمیق به داده بسیار زیاد و با کیفیت، پهپادها می‌توانند نقش قابل توجهی در این زمینه داشته‌باشند.

در سال‌های اخیر شرکت‌های مختلفی با هدف تولید پهپاد تجاری برای مقاصد مختلف بوجود آمده است که موارد زیر جزو مطرح‌ترین آن‌ها می‌باشند:

- DJI
- Parrot
- Yuneec
- Kespry
- Autel Robotics
- Skydio
- Insitu
- Delair
- Ehang
- Aerialtronics

مانند تمام ابزارهای جدید، پهپادهای تجاری نیز دارای محدودیت‌های مخصوص خود می‌باشند. این محدودیت‌ها معمولاً در بخش نرم‌افزار (به‌طور دقیقتر ثابت‌افزار یا firmware) پهپادها می‌باشد. برای مثال محدودیتی به نام No Fly Zones (NFZ) در این نوع پهپادها وجود دارد که مانع از پرواز آن‌ها در مناطق محافظت‌شده (مانند حریم هوایی فرودگاه‌ها) می‌شود. البته برخی از این محدودیت‌ها لازم و ضروری هستند اما تعدادی از این موارد، محدودیت‌هایی هستند که مانع از عملکرد کامل و دقیق پهپاد شود. درنتیجه مباحث سیستمی، نرم‌افزاری، سخت‌افزاری و تکنیک‌ها هک و مهندسی معکوس، برای ایجاد تغییرات و برداشتن این نوع محدودیت‌ها استفاده می‌گردند.
همچنین با توجه به وجود انجمن (forum) و community های مختلف در سطح اینترنت و فعالیت عده کثیری از کاربران حرفه‌ای در این زمینه، آموزش‌های بسیاری درزمینه نحوه هک و دور زدن محدودیت‌ها در سطح اینترنت وجود دارد. همچنین تعدادی نرم‌افزار اختصاصی نیز برای این منظور تهیه‌شده است. انجمن‌ها، منابع و نرم‌افزارهای مخصوص این کار (مخصوص پهپادهای DJI) به‌صورت زیر می‌باشند:

- فروم رسمی DJI
- فروم MavicPilots
- فروم phantomhelp
- ریپازیتوری dji-firmware-tools
- ریپازیتوری DUMLdore

در این پروژه قصد داریم تا با بررسی کامل ثابت‌افزار کنترلر پهپاد DJI Mavic 2 و اجرای آن برروی شبیه‌ساز پردازنده ARM، به تحلیل و آنالیز آن بپردازیم. هدف از این تحلیل بررسی کامل قابلیت‌ها و اپلیکیشن‌های موجود بر روی کنترلر پهپاد می‌باشد. با انجام این تحلیل، می‌توان تغییراتی را در کنترلر به وجود آورد و یا حتی می‌توان با استفاده از روش‌های مهندسی معکوس توانایی تولید کنترلر مشابه را به‌دست آورد.
همچنین علاوه بر تلاش برای اجرای ثابت‌افزار بر روی شبیه‌ساز، بخش‌های مختلف فایل دامپ ثابت‌افزار بررسی شد و با استفاده از ابزارهای مختلف (مانند binwalk و firmwalk) ابعاد دیگری از ثابت‌افزار مورد بررسی قرار گرفت. همچنین فایل jar موجود بر در یکی از دایرکتوری‌های ثابت‌افزار بررسی و دیکامپایل گردید.

۳.۱ چالش‌ها

با توجه به ماهیت پروژه طبیعتاً در اجرای پروژه، وجود چالش‌ها و سختی‌های مختلف اجتناب‌ناپذیر است. یکی از چالش‌هایی که در اجرای پروژه به آن برخوردیم، موجود نبودن اطلاعات کافی و دقیق در مورد خود سخت‌افزار و همچنین نحوه اجرای ثابت‌افزار روی سخت‌افزار مشابه بود. با توجه به تحقیقات و جستجوهایی که انجام شد، اکثر بحث‌های گروه‌ها و کامیونیتی‌ها حول محور ایجاد تغییر در ثابت‌افزار صرفاً با استفاده از یک سری روش‌های بسیار ساده بود و بحثی در مورد نحوه اجرا و تحلیل ثابت‌افزار یافته نشد.

البته تعدادی سؤال در این فروم‌ها در مورد ابهامات و نحوه ایجاد شبیه‌ساز از چند نفر از افراد باتجربه پرسیده شد اما تاکنون پاسخ خاصی دریافت نشده است. همچنین یکی از پیش‌نیازهای اعمال برخی از تغییرات در ثابت‌افزار پهپاد، داشتن خود پهپاد بود و بدون داشتن پهپاد نمی‌توان به هک برخی از قسمت‌های آن پرداخت.

۴.۱ کاربردها

نتیجه این پروژه کاربردهای مختلفی می‌تواند داشته باشد. برای مثال با استفاده از روش‌های مهندسی معکوس می‌توان نحوه ارتباط میان کنترلر و پهپاد را بدست آورد و همچنین ساختار کلی کنترلر را نیز کشف کرد. پس از آن نیز می‌توان کنترلر شخصی سازی شده برای کنترل کردن پهپاد ایجاد کرد.

۲ تحقیقات مقدماتی

۱.۲ تحقیق درباره نوع قطعات بورد اصلی و بورد ریموت‌کنترلر

یک نمونه از محصول مورد بررسی در اختیارمان بود امّا باز کردن آن به دلایل مختلف امکان‌پذیر نبود. برای مثال مطمئن نیستیم که آیا باز کردن دستگاه موجب فعال شدن قابلیت خودتخریبی (در صورت وجود) می‌شود یا خیر. بنابراین به جستجو در فضای نت پرداختیم. مدل دستگاه را جستجو کرده و تصاویر و بلاگ‌های مربوطه را مطالعه کردیم. در نتیجه‌ی این جستجوها به چند تصویر که مدل و نوع ماژول‌های روی بورد در آن‌ها مشخص است و یک مخزن گیت‌هاب رسیدیم که تا حدودی به توضیح و تشریح ماژول‌های مورد استفاده در این محصول پرداخته است. متاسفانه تصویر قابل استفاده‌ای از بورد ریموت‌کنترلر نیافتیم و همچنین مخزن مذکور اطلاعاتش تکمیل نیست.

۱.۱.۲ برد اصلی پهپاد

برد اصلی پهپاد mavic pro ۲ که با نام WM۲۴۰ شناخته می‌شود، وظیفه پردازش، ذخیره‌سازی و ارسال اطلاعات را دارد. این برد شامل بخش‌های مختلفی از جمله encoder video (هم برای FPV و هم برای SD-Card)، VPS (یا video recognition for positioning) و کنترل gesture می‌باشد. همچنین برد دارای فرستنده و گیرنده (Transceiver) برای ارتباطات رادیویی می‌باشد.
اجزای اصلی برد پهپاد موارد زیر می‌باشند.

- **پردازنده H3**
پردازنده Allwinner H3 (sun8iw7p1) وظایف انکود کردن ویدیو، پردازش مباحث vision پهپاد و کنترل پرواز را بر عهده دارد.

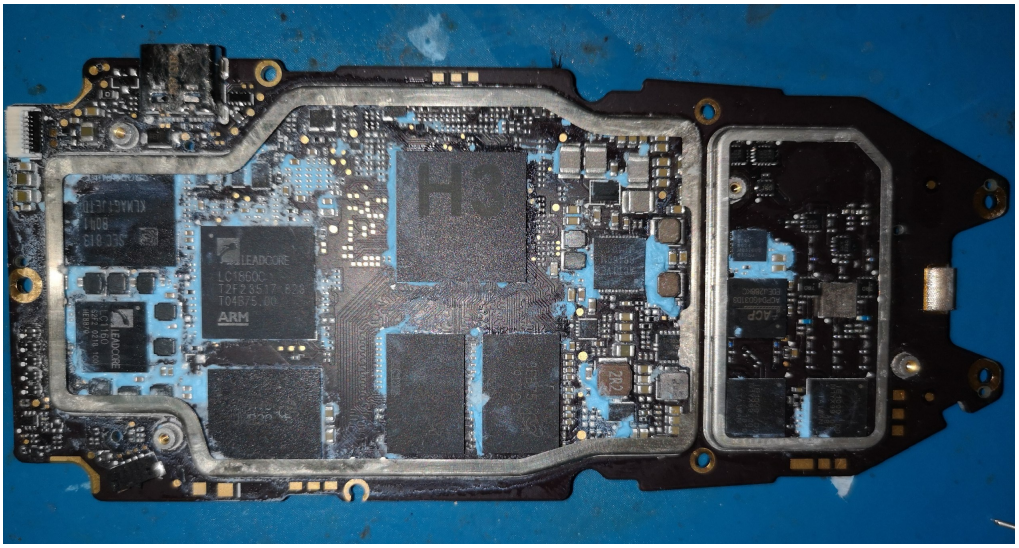
- **پردازنده LC1860C**
پردازنده Leadcore LC1860C وظایف ارتباطات رادیویی و و کنترل پرواز هوشمند را بر عهده دارد.

- **آی سی LC1160**
از آی سی برای تامین ولتاژ و جریان در مسیر اجرای دستور، و همچنین برای حفاظت در مدار و تامین انرژی آی سی ها استفاده می‌شود. از Leadcore LC1160 به عنوان آی سی در برد پهپاد استفاده می‌شود.

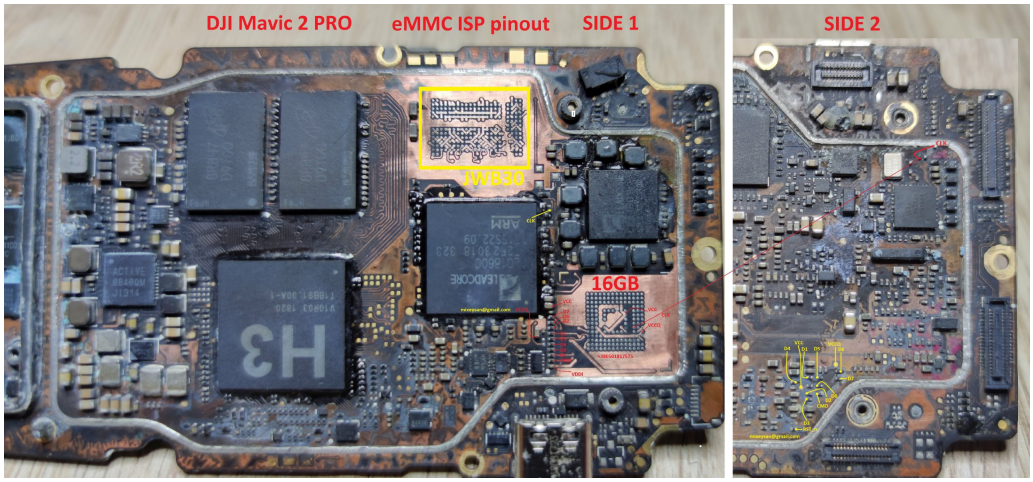
- **RAM**
از ماژول Micron MT۲۹TZZZ۴D۴BKERL-۱۲۵ به عنوان حافظه اصلی استفاده می‌شود. طبق بررسی ، این برد دارای ۸ گیگابایت DDR4 SDRAM می‌باشد.

- **حافظه**
از ماژول Samsung KLMAG1JETD-B041 به عنوان حافظه جانبی پهپاد استفاده می‌شود. البته لازم به ذکر است که می‌توان از SD Card نیز در پهپاد استفاده کرد.

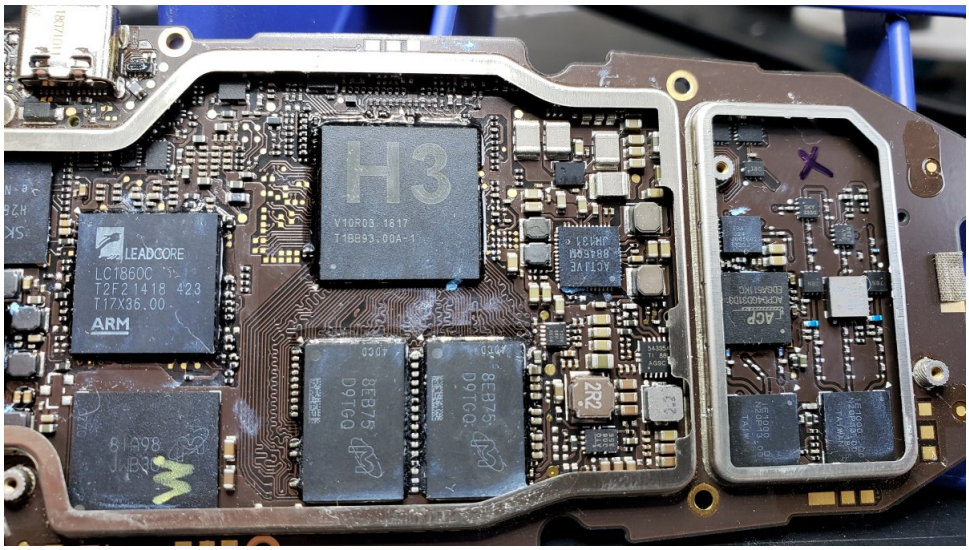
شکل ۱: برد اصلی پهباد پیش از جداسازی محافظ فلزی ـ منبع



شکل ۲: برد اصلی پهباد پس از جداسازی محافظ فلزی ـ منبع



شکل ۳: پشت و روی بورد اصلی پهباد



شکل ۴: تصویری واضح از بورد اصلی پهباد و قطعات و ماژول‌های مورد استفاده ـ منبع

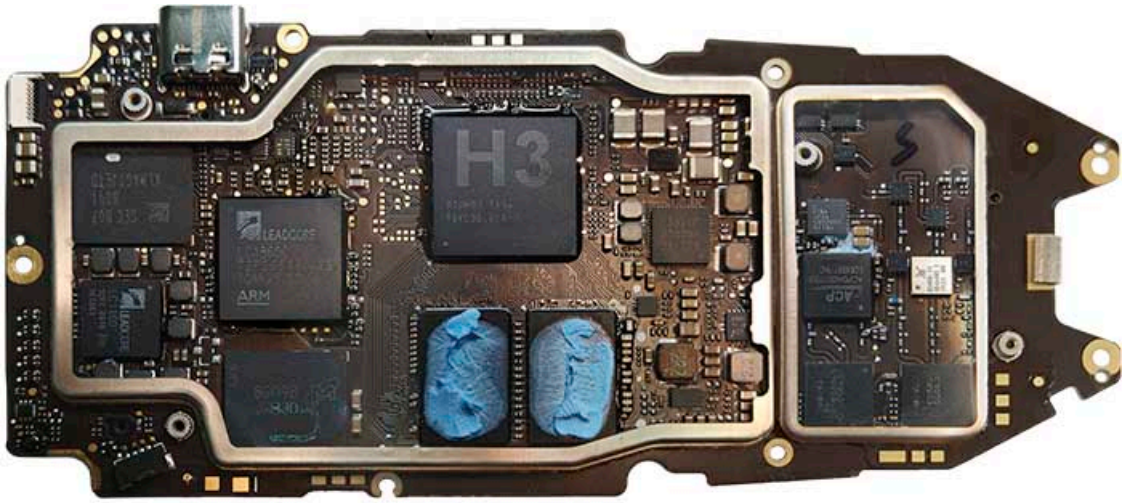
- **Active-Semi Advanced PMIC**
مدار مجتمع مدیریت نیرو (PMIC) ACTIVE 8846QM برای مدیریت اپلیکیشن‌های چندهسته‌ای پهپاد استفاده می‌گردد.

- **مودم**
از مودم LTE با نام IRIS۴۱۱ ACP در پهپاد استفاده می‌شود.

- **ماژول gps**

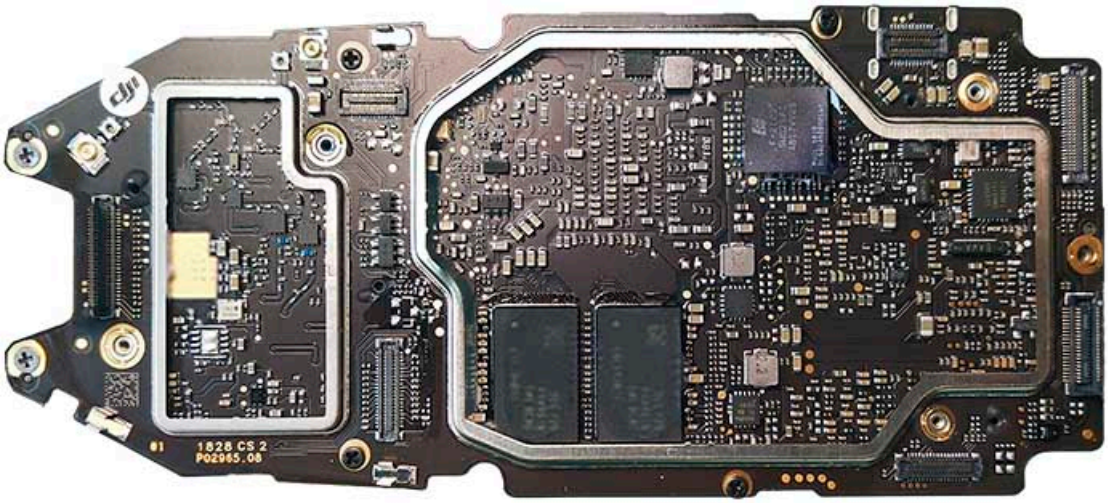
- **بورد سنسور دوربین**

- **باتری**



شکل ۵: تصویر جلوی بورد اصلی پهپاد

شکل ۶: پشت برد اصلی پهباد پیش از جداسازی محافظ فلزی



شکل ۷: پشت برد اصلی پهباد پس از جداسازی محافظ فلزی

۲.۱.۲ لینک‌های مفید

- ریپازیتوری مرجع برای ابزارهای مرتبط با ثابت‌افزار DJI

- صفحه اطلاعات برد اصلی پهپاد
- توضیحات در مورد کنترلر پهپاد mavic 2 prop

۲.۲ تحقیق راجع به نسخه اندروید مورد استفاده در ثابت‌افزار

به جهت راه‌اندازی سامانه بر روی شبیه‌ساز به سراغ یافتن نسخه اندروید مورد استفاده از کوادکوپتر رفتیم. برای این منظور ابتدا با جستجو در فضای اینترنت به نتیجه‌ی مشخصی نرسیدیم اما سپس با بررسی فایل‌های موجود در فایل‌سیستم، به سندی تحت عنوان build.prop برخوردیم که اطلاعاتی راجع به کوادکوپتر در داخلش نوشته شده است. این فایل در عموم سیستم‌های اندرویدی وجود دارد و به کمک آن می‌توان ویژگی‌هایی از سیستم را تنظیم کرد. پارامترهای مهم موجود در این فایل در ثابت‌افزار مورد بررسی ما موارد زیر هستند:

```
ro.build.display.id=leadcore1860
ro.build.version.sdk=19
ro.build.version.release=4.4.4
ro.product.model=L1860
ro.product.brand=Leadcore
ro.product.name=full_wm240_dz_rp0010_v1
ro.product.cpu.abi=armeabi-v7a
net.bt.name=Android
```

همانطور که پیداست، اا‌ز اندروید نسخه ۴.۴.۴ استفاده شده است.

۳.۲ تحقیق راجع به فایل‌سیستم اندروید

از آنجا که اندروید فریم‌ورکی [۱] برای لینوکس است، ساختار فایل‌سیستم[۲] آن هم شباهت زیادی به فایل‌سیستم لینوکس دارد. با این حال فایل‌سیستم اندروید ویژگی‌هایی مختص به خود و متفاوت با لینوکس را هم داراست.

در فایل‌سیستم اندروید به طول معمول ۶ پارتیشن اصلی وجود دارد.

- \boot
- \system
- \recovery
- \data

[۱] Framework
[۲] File system

- \cache
- \misc
- { \sdcard }
- { \sd-ext }

۴.۲ بررسی و جستجوی انجمن‌ها، فروم‌ها، و ریپازیتوری مرتبط با پهپادهای DJI و بررسی ابزارهای مختلف

جهت یافتن اطلاعات بیشتر در مورد نحوه اجرای پروژه و همچنین اطلاعات دقیق در مورد سخت‌افزار پروژه، سایت‌ها، فروم‌ها و ریپازیتوری‌های مختلف بسیاری در مورد پهپادهای DJI بررسی شد. برای نمونه فروم‌های اصلی DJI و mavicpilots بررسی شد و همچنین یک سری سؤال در مورد نحوه اجرای ثابت‌افزار پرسیده شد. همچنین چندین ریپازیتوری مختلف در سایت گیت‌هاب موردبررسی قرار گرفت. در ادامه لینک چندین منبع و بحث در کامیونیتی‌های مختلف در مورد این موضوع قرار داده‌شده است:

- بحث در مورد نحوه هک کردن ویژگی‌های مختلف پهپاد Mavic 2
- بحث در مورد نحوه هک کردن مجوز FCC پهپادهای Mavic ۲
- پرسش در مورد نحوه اجرای ثابت‌افزار بر روی شبیه‌ساز
- نکات مرتبط با پهپادهای Mavic
- صفحه اصلی روش‌های هک پهپادهای DJI
- ریپازیتوری DUMLdore برای هک و ایجاد تغییرات در پهپاد
- ریپازیتوری DJI Firmware Tools، منبع جامع و کامل برای ابزارهای مدیریت ثابت‌افزار DJI

۳ شبیه‌ساز

۱.۳ تحقیق در مورد شبیه‌سازهای پردازنده ARM

به منظور اجرا و بازسازی کردن شرایط واقعی اجرای برنامه نیازمند یک شبیه‌ساز هستیم که با توجه به بورد و قطعات مورد استفاده در پهباد نیازمند یک شبیه‌ساز پردازنده ARM هستیم.

شبیه‌سازهای نرم‌افزاری ARM به شما این امکان را می‌دهند که یک دستگاه ARM شبیه‌سازی شده را بر روی سیستم اصلی کامپیوتر خود اجرا کنید، خواه ویندوز، لینوکس یا هر سیستم عامل دیگری باشد. این به شما امکان توسعه و آزمایش نرم‌افزار را بر روی سیستم شخصی خودتان می‌دهد و می‌توانید در زمان نیاز، آن را به یک دستگاه واقعی ARM منتقل کنید.

در صورت ناموفق بودن این تلاش، اقدام بعدی می‌تواند خرید قطعات اصلی و اجرای کد مستقیما بر روی قطعات فیزیکی باشد. مشخص است که این گزینه هزینه بیشتری را در بر دارد.

به منظور انتخاب شبیه‌ساز مناسب چندین گزینه پیش رو داریم.

- ARMware
- Microsoft Device Emulator 3.0 (Standalone Release)
- Softgun - the Software ARM
- QEMU CPU Emulator
- SkyEye

از میان این شبیه‌سازها بهترین انتخاب از نظر سادگی در استفاده و جامع بودن شبیه‌ساز QEMU است. چرا که سایر شبیه‌سازها از طیف گسترده‌ای از پردازنده‌های پشتیبانی نمی‌کنند و مستندات و آموزش‌های موجود از آن‌ها هم به اندازه QEMU جامع و کامل نیست. بنابر دلایل بالا ما در انجام پروژه خود از شبیه‌ساز QEMU استفاده کردیم.

- معرفی شبیه‌سازهای آزاد ARM

۲.۳ بررسی و استفاده از QEMU برای شبیه‌سازی سیستم‌عامل‌های مختلف

برای آشنایی بیشتر با شبیه‌ساز QEMU و بررسی نحوه ایجاد محیط مجازی، چندین آزمایش کلی در این زمینه انجام شد، که نحوه اجرای آن‌ها در این بخش توضیح داده می‌شوند.

۱.۲.۳ آزمایش ۱: شبیه‌سازی اندروید 8.1 نسخه پردازنده x86

در ابتدا لازم است که فایل نصبی اندروید ۱.۸ دانلود شود. با استفاده از این لینک می‌توان این کار را انجام داد.
در ابتدا لازم است که با استفاده از کامند زیر، یک Drive Hard مجازی ایجاد شود:

```
$ qemu-img create -f qcow2 androidx86_hda.img 10G
```

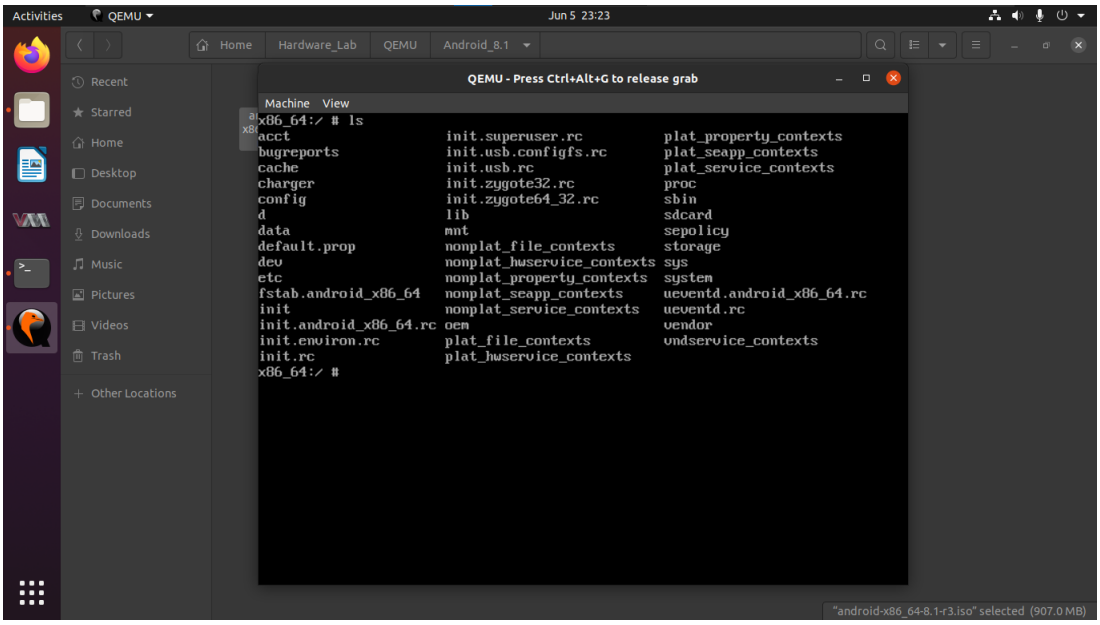
سپس با استفاده از دستور زیر، فایل نصبی را اجرا می‌کنیم.

```
$ qemu-system-x86_64 \
-m 2048 \
-smp 2 \
-soundhw es1370 \
-device virtio-mouse-pci -device virtio-keyboard-pci \
-serial mon:stdio \
-boot menu=on \
-net nic \
-net user,hostfwd=tcp::5555-:22 \
-device virtio-vga,virgl=on \
-display gtk,gl=on \
-hda androidx86_hda.img \
-cdrom android-x86_64-1.8-r3.iso
```

با استفاده از آموزش موجود در لینک به نصب اندروید می‌پردازیم. البته می‌توان به طور مستقیم و بدون نصب، به اجرای سیستم‌عامل پرداخت.
بعد از نصب کامل سیستم‌عامل اندروید، با استفاده از کامند زیر می‌توان سیستم عامل را اجرا کرد.

```
$ qemu-system-x86_64 \
-m 2048 \
-smp 2 \
-soundhw es1370 \
-device virtio-mouse-pci -device virtio-keyboard-pci \
-serial mon:stdio \
-boot menu=on \
-net nic \
-net user,hostfwd=tcp::5555-:22 \
-device virtio-vga,virgl=on \
-display gtk,gl=on \
-hda androidx86_hda.img
```

در شکل زیر ترمینال اندروید 8.1 را مشاهده می‌نمایید. با توجه به اینکه هدف این آزمایش صرفا آشنایی با ساختار QEMU و نحوه اجرای یک سیستم عامل با این شبیه‌ساز است، صرفا اجرای اندروید بر روی ترمینال کافی بود.



شکل ۸: ترمینال اندروید 8.1 اجرا شده بر روی QEMU

۲.۲.۳ آزمایش ۲: شبیه‌سازی rpios lite نسخه پردازنده ARM

در آزمایش دوم به نصب Os Pi Raspberry نسخه lite مخصوص پردازنده‌های آرم پرداختیم.
با فرض نصب بودن QEMU، در ابتدا لازم است که فایل سیستم‌عامل دانلود شود. با استفاده از دستور زیر به دانلود نسخه مشخصی از سیستم‌عامل rpi می‌پردازیم.

```
$ wget https://downloads.raspberrypi.org/raspios_lite_armhf/images/
raspios_lite_armhf-2021-01-12/2021-01-11-raspios-buster-armhf-lite.zip
```

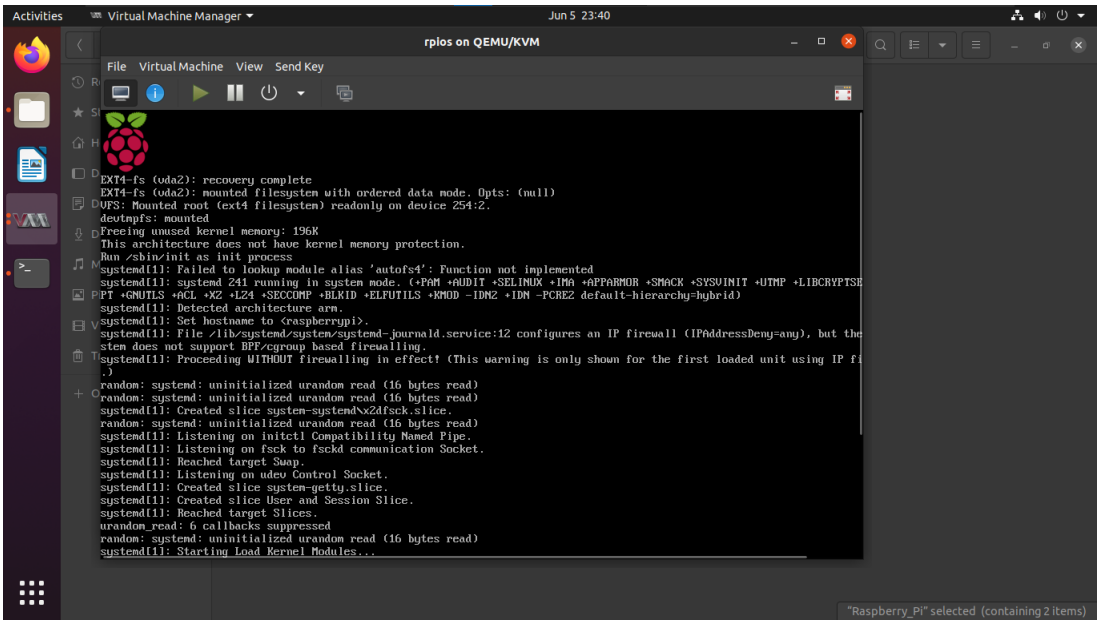
پس از آن، لازم است که کرنل‌های مورد نیاز این سیستم‌عامل از طریق این ریپازیتوری دانلود شود.

```
$ git clone https://github.com/dhruvvyas90/qemu-rpi-kernel
```

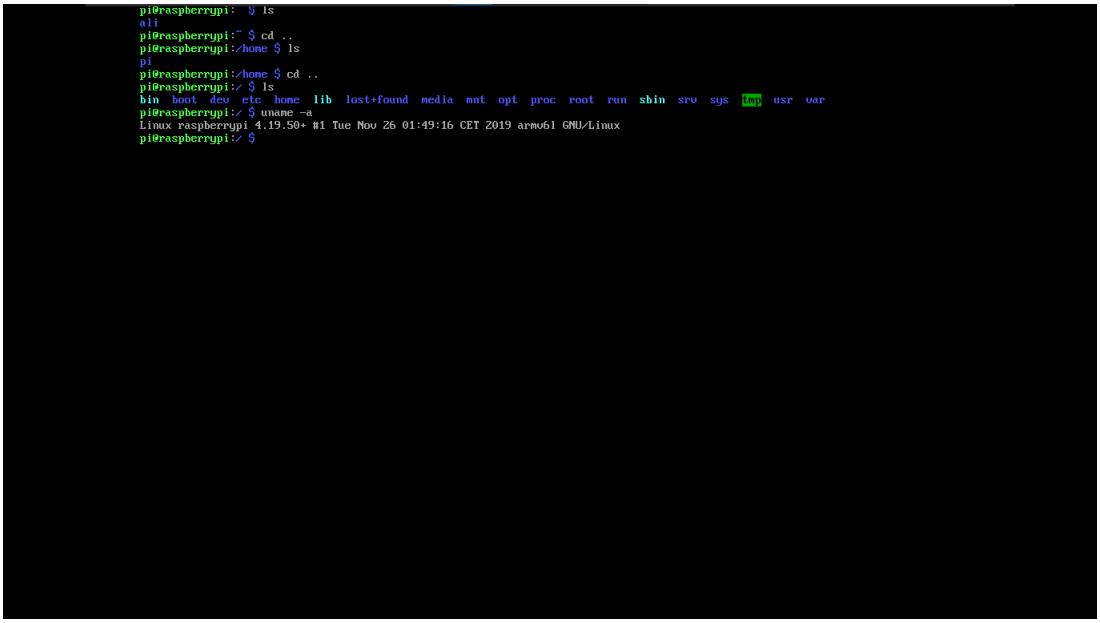
اکنون با استفاده از کامند زیر، سیستم عامل Lite Pi Raspberry را بالا اجرا می‌کنیم:

```
$ sudo virt-install \
--name rpios  \
--arch armv6l \
```

```
۴  --machine versatilepb \
۵  --cpu arm1176 \
۶  --vcpus 1 \
۷  --memory 256 \
۸  --import  \
۹  --disk 2021-01-11-raspios-buster-armhf-lite.img,format=raw,bus=virtio \
۱۰ --network bridge,source=virbr0,model=virtio  \
۱۱ --video vga  \
۱۲ --graphics spice \
۱۳ --boot 'panic=1 ,kernel_args=root=/dev/vda2,kernel=qemu-rpi-kernel/kernel-qemu-4.50
۱۴ --events on_reboot=destroy
```



شکل ۹: لود شدن سیستم‌عامل rpi os بر روی QEMU



شکل ۱۰: ترمینال rpi os اجرا شده بر روی QEMU

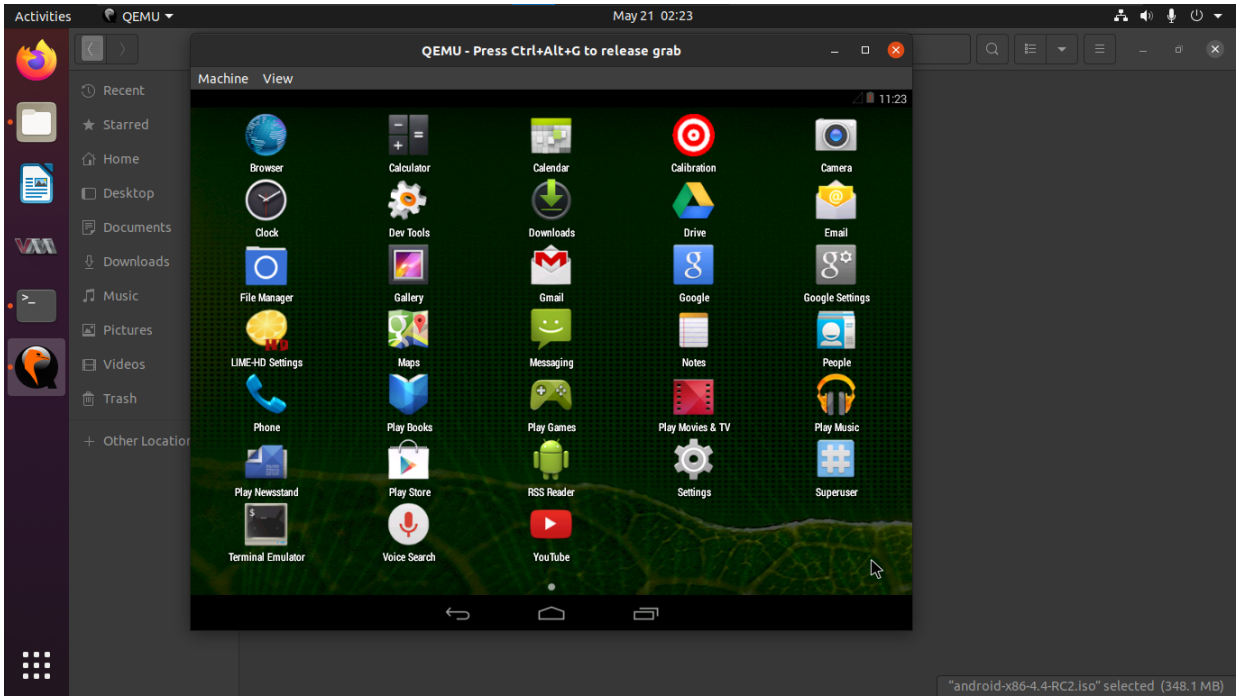
۳.۲.۳ آزمایش ۳: شبیه‌سازی اندروید 4.4 نسخه پردازنده x86

در این آزمایش قصد داریم تا نسخه ۴.۴ اندروید مخصوص پردازنده‌های x۸۶ را نصب و اجرا کنیم. توجه داشته باشید که نسخه اندروید مورد استفاده در ثابت‌افزار کنترلر پهپاد، نسخه ۴.۴ اندروید می‌باشد و نصب و اجرای درست این سیستم‌عامل بر روی QEMU اهمیت بسیار زیادی دارد.
برای نصب، در ابتدا باید یک درایو مجازی ایجاد شود:

```
$ qemu-img create -f qcow2 android44_qcow2.img 8G
```

سپس لازم است که فایل اندروید از این لینک دانلود شود. پس از دانلود با اجرای این کامند، می‌توان به نصب یا اجرای اندروید پرداخت:

```
$ qemu-system-x86_64 \
-vga std -m 1024 \
-drive file=android44_qcow2.img,cache=none \
-soundhw es1370 \
-cdrom android-x86-4.4-RC2.iso
```



شکل ۱۱: محیط اجرای اندروید ۴.۴ بر روی QEMU

۴ اجرا

۱.۴ دیکامپایل کردن برنامه موجود در ثابت‌افزار و بدست‌آوردن کد اندروید

با گسترده‌سازی فایل m2rc-dump.zip وارد پوشه system/framework می‌شویم. فایلی به نام input.jar را به کمک ابزار jadx با دستور

```
./jadx address-to-project/system/framework
```

دیکامپایل می‌کنیم و کد منبع به زبان جاوا/اندروید به دست می‌آید.
کنترلر به این صورت کار می‌کند که انواع عملیات‌هایی که می‌شود با دسته انجام داد و دکمه‌هایی که می‌شود کلیک کرد را به کمک کتابخانه‌های اندروید inject می‌کند (ارسال به کوادکوپتر)

- ابزار آنلاین برای decompile کردن کد جاوا

۲.۴ تحلیل firmware با ابزار firmwalk

تجزیه و تحلیل دستی ثابت افزار نیازمند دقت و زمان زیادی است و به دلیل محدودیت های زمانی اغلب نمی‌توان آنالیز دستی انجام داد. بنابراین، تجزیه و تحلیل خودکار ثابت افزار بسیار کمک کننده خواهد بود. ما برای این کار از ابزار firmwalker استفاده کردیم که ابزاری برای تحلیل firmware است و از آدرس https://github.com/craigz۲۸/Firmwalker قابل دسترسیست. فرمواکر در واقع یک script bash است که موارد زیر را شناسایی می‌کند.

```
etc/shadow and etc/passwd
list out the etc/ssl directory
search for SSL related files such as .pem, .crt, etc.
search for configuration files
look for script files
search for other .bin files
look for keywords such as admin, password, remote, etc.
search for common web servers used on IoT devices
search for common binaries such as ssh, tftp, dropbear, etc.
search for URLs, email addresses, and IP addresses
Experimental support for making calls to the Shodan API using the Shodan CLI
```

برای استفاده کافی است ثابت افزار را از حالت فشرده خارج کنیم (برای نتیجه مطمئن‌تر می‌توان از binwalk برای گسترده سازی استفاده کرد) و آن را در مسیر firmwalker/firmwalker-master قرار دهیم و همان جا دستور زیر را اجرا کنیم. در نتیجه‌ی تحلیل یک فایل متنی به فرمت txt. در همان آدرس نوشته خواهد شد. که شامل اطلاعات مذکور است.

```
$ ./firmwalker.sh firmwareName.extracted
```

۳.۴ اجرای ثابت‌افزار مربوط به کنترلر بر روی QEMU

در این بخش، برای تحلیل بهتر و مناسبتر ثابت‌افزار کنترلر، قصد داریم تا با آزمایش‌های مختلف، به اجرای این ثابت‌افزار بر روی شبیه‌ساز بپردازیم. در ابتدا با ساختار کلی فایل نصبی اندروید آشنا می‌شویم. سپس فایل system.sfs که یکی از مهم‌ترین فایل‌های موجود در iso اندروید می‌باشد را بررسی می‌کنیم. و درنهایت سه آزمایش مختلف جهت اجرای ثابت افزار انجام می‌دهیم.

۱.۳.۴ ساختار کلی فایل نصبی اندروید

ساختار فایل نصبی iso اندروید به شکل زیر می‌باشد:

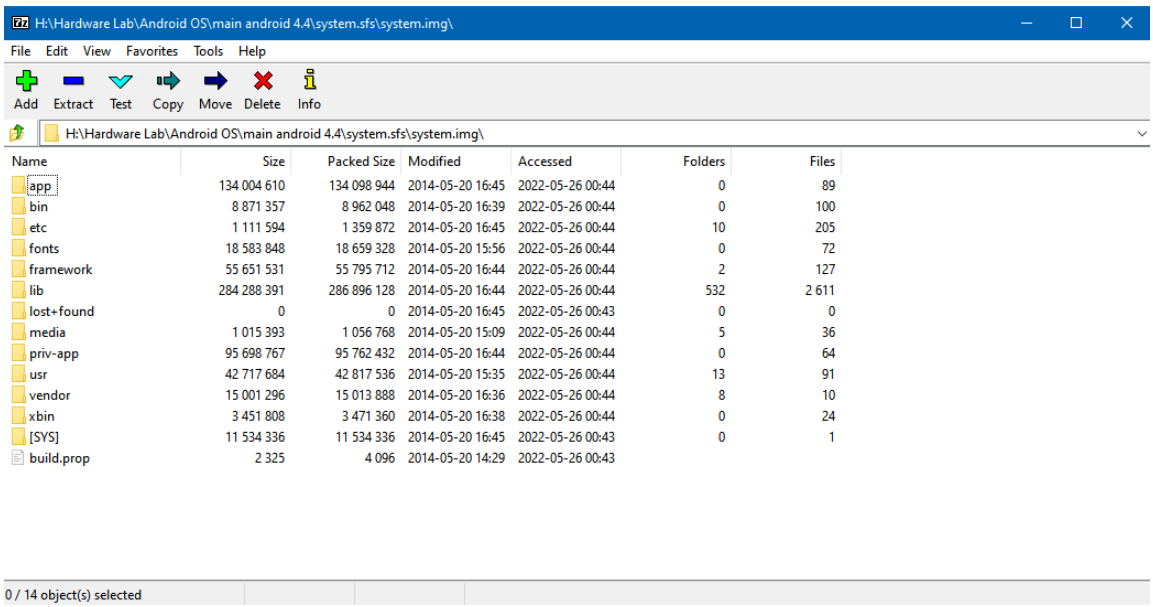
```
/.disk
/boot
/efi
/isolinux
/[BOOT]
initrd.img
install.img
kernel
ramdisk.img
system.sfs
TRANS.TBL
```

توجه داشته باشید که ممکن است فایل‌ها و یا دایرکتوری‌های دیگری با توجه به نسخه اندروید در فایل iso موجود باشد و یا برخی از فایل‌ها و دایرکتوری‌ها در iso قرار نگرفته باشند.
از بین دایرکتوری‌ها و فایل‌های موجود، دایرکتوری boot و فایل system.sfs و همچنین kernel اهمیت بیشتری نسبت به بقیه دارند. در دایرکتوری boot کانفیگ‌های مربوط به بوت لینوکس قرار دارد و عملا بدون این دایرکتوری، سیستم‌عامل اجرا نمی‌شود.
kernel نیز هسته و اصلی‌ترین بخش سیستم‌عامل است که پایه‌ای ترین سرویس‌ها برای سایر بخش‌های سیستم‌عامل را مهیا می‌کند.
فایل system.sfs نیز که یک فایل فشرده می‌باشد، شامل فایل‌ها و دایرکتوری‌های اصلی سیستم‌عامل است.
برای اجرای ثابت‌افزار، لازم است که به گونه‌ای فایل‌ها و دایرکتوری‌های آن را به صورت system.sfs در بیاوریم و آن را کنار سایر فایل‌های مورد نیاز در ،iso قرار دهیم و فایل نصبی جدید را ایجاد کنیم. در ادامه چگونگی انجام این مراحل را بررسی می‌کنیم.

۲.۳.۴ نحوه ایجاد فایل system.sfs

ساختار فایل system.sfs به اینگونه است که در داخل آن یک فایل فشرده با نام system.img وجود دارد که داخل این فایل، فایل سیستم اصلی سیستم‌عامل قرار دارد. نکته قابل توجه این است که در فایل دامپی که

ما برای ثابت‌افزار در اختیار داریم، شامل یک دایرکتوری با نام system است که فایل سیستم اصلی ثابت‌افزار را دارا می‌باشد و با تبدیل این دایرکتوری به system.img و سپس system.sfs، این دایرکتوری را آماده اجرا خواهیم کرد.



شکل ۱۲: ساختار داخلی system.sfs

برای ایجاد فایل img، از نرم‌افزار ImgBurn استفاده می‌کنیم و برای ایجاد فایل فشرده sfs از دو روش می‌توان استفاده کرد:

- استفاده از ابزار RMXTools v1.5 و بر اساس راهنمایی این ویدیو
- با استفاده از ابزار SquashFS اوبونتو و کامند mksquashfs

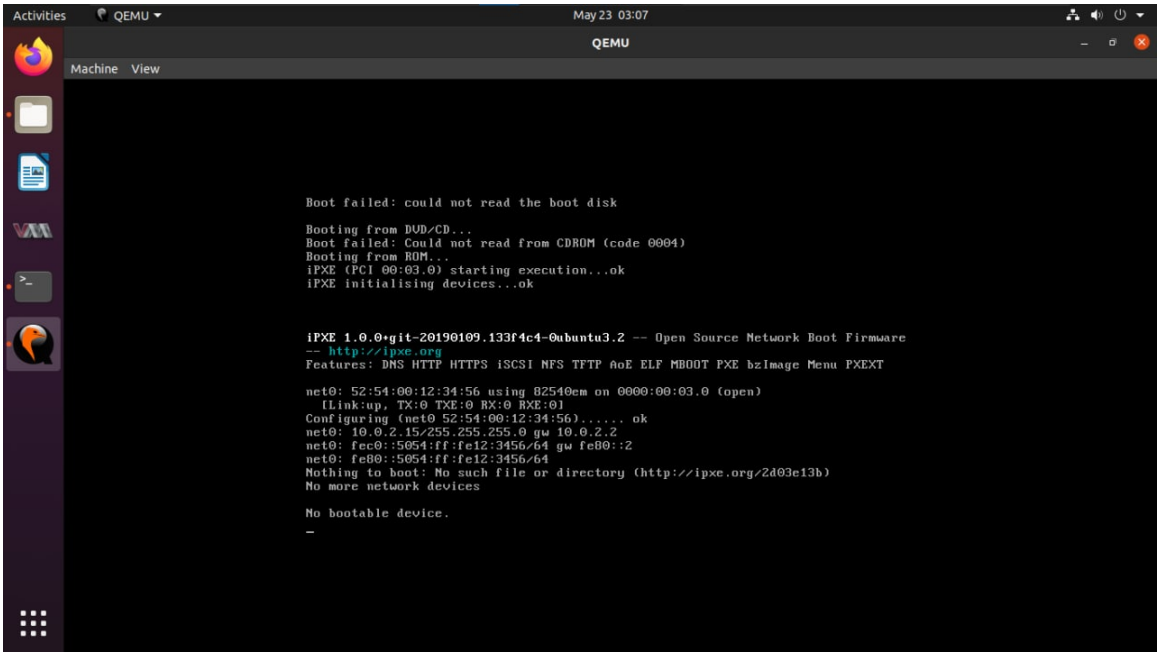
در ادامه کار، با انجام چند آزمایش قصد داریم تا اجرای ثابت‌افزار را شبیه‌سازی کنیم.

۳.۳.۴ آزمایش ۱: شبیه‌سازی اندروید 4.4 نسخه پردازنده x86

اجرای این نسخه اندروید شبیه اجرای اندروید 8.1 است. صرفا لازم است که با استفاده از این لینک به دانلود فایل اندروید می‌پردازیم، سپس با استفاده از کامند زیر، به اجرای سیستم‌عامل مانند بخش ۳.۲.۳ می‌پردازیم.

۴.۳.۴ آزمایش ۲: شبیه‌سازی نسخه دستکاری شده اندروید 4.4 با استفاده از فایل‌های دامپ

در ابتدا فایل‌ها و دایرکتوری‌های موجود در دامپ ثابت‌افزار کنترلر را به system.sfs تبدیل می‌کنیم و فایل sfs جدید را جایگزین system.sfs قبل در فایل نصبی اندروید قرار می‌دهیم. سپس به اجرای اندروید تغییر داده شده می‌پردازیم.

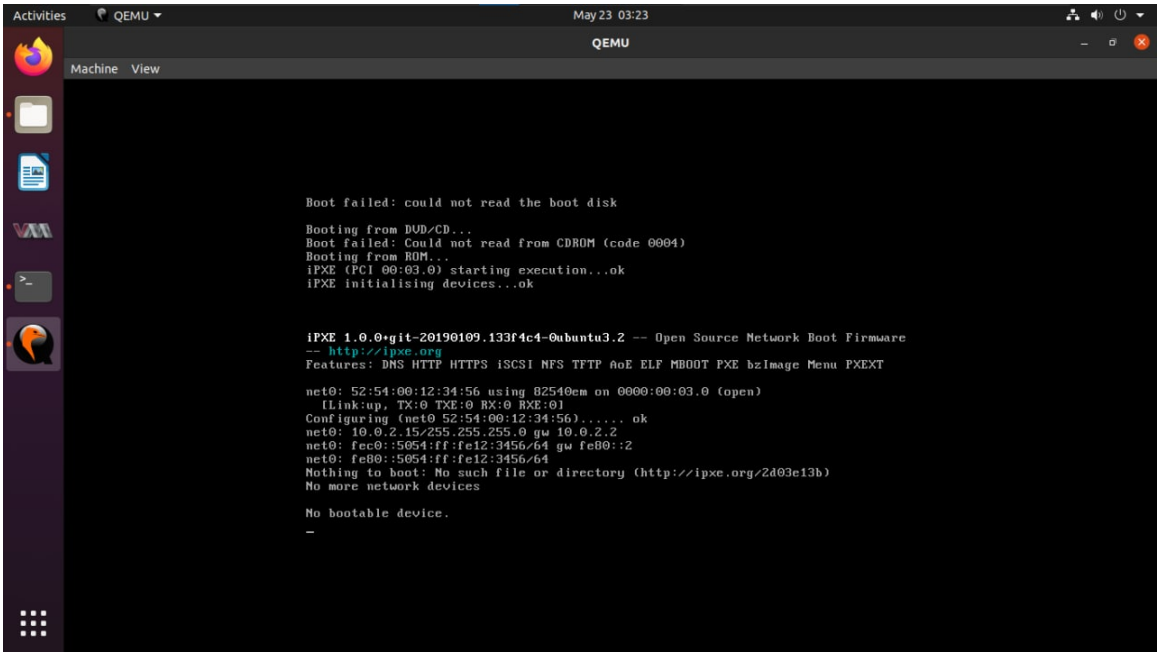


شکل ۱۳: نتیجه اجرای آزمایش ۲

همانگونه که در تصویر ۱۳ مشاهده می‌کنید، اجرای این آزمایش به خطا می‌خورد. علت خطا نیز این است که سیستم با استفاده از CD/DVD بوت نمی‌شود و سیستم نمی‌تواند اجرا شود. توجه داشته باشید که با هر دو روش ارائه شده در بخش قبل به با هردو روش ذکر شده، فایل sfs ایجاد می‌گردد، اما در هر دو روش، به مشکل یکسان برخورد می‌کنیم.

۵.۳.۴ آزمایش ۳: شبیه‌سازی نسخه دستکاری شده اندروید 4.4 با استفاده از ترکیب فایل‌های دامپ و اندروید

برای این آزمایش، فایل‌ها و دایرکتوری‌های موجود در دامپ به صورت replace در دایرکتوری system اندروید ۴.۴ کپی می‌کنیم و سپس فایل system.sfs را در محتویات فایل iso ایجاد می‌کنیم و سپس فایل نصبی جدید را ایجاد و اجرا می‌نماییم.



شکل ۱۴: نتیجه اجرای آزمایش ۳

در اجرای این قسمت نیز با مشکل بخش قبل مواجه می‌شویم.

۵ نتیجه‌گیری

در این پروژه تلاش شد تا با انجام تحقیقات درباره اجزای پهپاد و ساختار فایل ثابت‌افزار و همچنین تحلیل و بررسی بخش‌های مختلف ثابت‌افزار به همراه اجرای آن بر روی شبیه‌ساز، عملکرد ثابت‌افزار کنترلر و کوادکوپتر پهپاد تجاری DJI Mavic Pro 2 بررسی شود.
در تحقیقات انجام شده، سخت‌افزار پهپاد و کنترلر به طور کامل بررسی شد و همچنین فایل‌سیستم اندروید بررسی گردید. سپس لینک‌ها، ریپازیتوری‌ها و منابع مفید معرفی گردید. در بخش شبیه‌ساز نیز، در مورد شبیه‌سازهای مختلف پردازنده ARM بررسی و تحقیق انجام گردید و آزمایش‌های مختلفی برای آشنایی بیشتر با شبیه‌ساز QEMU ترتیب داده شد.
در بخش اجرا نیز سعی شد تا با استفاده از اطلاعات و ابزارهایی که در اختیار است، به اجرا و تحلیل بیشتر ثابت‌افزار بپردازیم. برای راحتی کار نیز، تنها ثابت افزار مربوط به کنترلر پهپاد بررسی گردید. در نهایت نیز با توجه به برخی از مشکلات و چالش‌ها، اجرای ثابت‌افزار بر روی شبیه‌ساز با مشکل مواجه شد و نتوانستیم شبیه‌ساز را به طور کامل بر روی شبیه‌ساز اجرا کنیم. البته نکته قابل توجه این است که با سرچ و پرس‌جو در سطح اینترنت و فروم‌ها و کامیونیتی‌های مربوطه، کسی تاکنون اقدام به اجرای ثابت‌افزار بر روی شبیه‌ساز نکرده بود و راهنمای جامع و قابل‌اعتمادی برای انجام این کار وجود نداشت.
علاوه بر تلاش برای اجرای ثابت‌افزار بر روی شبیه‌ساز، بخش‌های مختلف دامپ ثابت‌افزار بررسی گردید. یکی از این موارد، بررسی فایل jar موجود در یکی از دایرکتوری‌های دامپ بررسی گردید و با استفاده از ابزار jadx این فایل دیکامپایل گردید و کد جاوای مرتبط با عملیات‌های قابل انجام با دکمه‌های کنترلر استخراج گردید. همچنین با استفاده از ابزار ،binwalk بخش‌های مرتبط دامپ بررسی و تحلیل گردید.
گام‌های بعدی برای بررسی و تحلیل ثابت‌افزار، استفاده از روش‌های دیگر و جهت‌های دیگر برای بررسی فایل‌های ثابت افزار است.